For Flux Sake: The Confluence of Socially- and Biologically-Inspired Computing for Engineering Change in Open Systems

Jeremy Pitt
Dept. of Electrical and Electronic Eng.
Imperial College London
London, SW7 2BT, England
Email: j.pitt@imperial.ac.uk

Emma Hart
Dept. of Computing
Edinburgh Napier University
Edinburgh, EH10 5DT, Scotland
Email: e.hart@napier.ac.uk

***Abstract*—This *position paper* is concerned with the challenge of engineering multi-scale and long-lasting systems, whose operation is regulated by sets of mutually-agreed, conventional rules. The core of the problem is that there are multiple, inter-dependent *dimensions of flux*, with numerous contextual factors to take into account. These dimensions of flux include, on the one hand, the set of rules itself; and on the other, the system components (population), their social network, and the operating environment. However, there appears to be no 'one size fits all' optimum ruleset for all combinations of population, social network and environment; nor (given the contextual factors) is there a planning-type algorithm that can compute an 'ideal' ruleset for any particular combination of population, social network and environment. These features of the problem suggest that recent advances in machine learning and evolutionary computation can provide the instruments for facilitating self-adaptation of a rule-based system over different timescales. This paper proposes that the integration of concepts from socially- and biologically-inspired computing can pave the way for eventual development of a computational framework that will enable principled (methodological) development of sustainable adaptive rule-based systems.**

1. Introduction

As information technology in the form of distributed devices and virtual agents becomes more and more embedded in our society, its architecture and deployment have begun to mirror human structures, whereby devices and agents form electronic "institutions" and virtual "organisations" which are governed by rules and policies. Examples include both technical systems (sensor networks, robotic swarms) and socio-technical systems (e.g. smart-grids, intelligent urban transportation, participatory sensing applications). Engineering and operating these types of systems is challenging: hard-wired local rules can result in complex global behaviours; rules can be executed without any awareness of the global system consequences; systems can have no capacity to adapt to unfamiliar environmental dynamics. As a result, there remains a significant gap in knowledge as to how to *engineer* adaptive systems which remain sustainable over long periods of time and continue to fulfil their goals in the face of dynamic and unpredictable changes.

In purely *social* systems, the Nobel prize-winning scientist Ostrom [1] has outlined eight principles by which institutions (i.e. groups of people) that operate according to rules and structures can achieve sustainability. Separately, study of *biological* systems reveals mechanisms by which species and/or populations can adapt over multiple timescales in response to changes in their environment: learning mechanisms enable individual adaptations over the course of a lifetime while evolutionary mechanisms adapt populations over the course of many generations. We propose to combine computational aspects of both systems in order to develop a new framework and set of design principles for engineering sustainable, distributed, autonomous systems. The proposed method benefits from bringing together techniques which overcome their individual limitations. Ostrom's principles define the requirement for self-awareness within a system in order to be sustainable, but without the necessary mechanisms to operationalise this. Biological systems on the other hand have no goal beyond "survival (of the fittest)" but do have operational mechanisms to enable adaptation to environmental change. By bringing these fields together, we propose that it will become possible to understand the principles by which it is possible to design and operate multi-scale and long-lasting adaptive systems.

Therefore the aim of this *position paper* is to highlight the 'scientific push' and 'application pull' that warrants a deeper investigation into the confluence of the socially-inspired and biologically-inspired computing methods for the purposes of engineering sustainable socio-technical systems. The goal is to examine the relevant factors from the social and biological domains that will pave the way for eventual development of a computational framework that will enable principled development of such systems.

Accordingly, the paper is structured as follows. Section 2 provides a brief background on the respective scientific fields. In Section 3 we identify four inter-dependent, *di-*

mensions of flux that drive this work: this concerns the institutional ruleset, the system components (the population), their social network, and the operating environment. Furthermore, there are several complicating 'non-functional' contextual factors which need to be taken into account, as discussed in Section 4. Section 5 and Section 6 lay out the proposed solution from (respectively) from the perspectives of socially- and biologically-inspired computing. Section 7 concludes with some comments on developing a unifying computational framework based on the confluence of socially- and biologically-inspired computing, as a basis for engineering open, adaptive, rule-based systems which can 'cope' with the manifold dimensions (and contextual factors) of flux.

2. Background

The study of social and biological systems has provided both the scientific foundations for sub-fields of Artificial Intelligence, and the basis for engineering methodologies used to develop solutions to problems in computer science.

For example, in the study of biological systems, the process of evolution has been identified as a mechanism by which a species (or a population) can adapt its forms and functions, over multiple timescales in response to environmental changes, resulting in sustainable biodiversity. This has established the scientific foundations of genetic algorithms [2] and evolutionary computation [3]. These approaches provide a way of generating solutions to optimisation problems in which different algorithms have different performance characteristics on (even relatively "small") variations of the same problem [4], and have been made systematic in the methodology of biologically-inspired computing [5].

Correspondingly, in the study of social systems, self-governing institutions have been proposed [1] as a means by which a group of people can adapt conventional rules and organisational structures, over multiple timescales in response to environmental changes, to achieve sustainable common-pool resource management. This has led to the ideas of norm-governed multi-agent systems [6] and electronic institutions [7]. These approaches offer, for example, a way of generating solutions to co-operative problem solving in which knowledge, (institutionalised) power and functionality are distributed between autonomous and heterogenous components [8], and have been made equally systematic in the methodology of socially-inspired computing [9].[1]

Both biologically-inspired and socially-inspired approaches have been applied to resource allocation, for example in job-shop scheduling [10] and in energy distribution in community energy system [11]. However, there has, to date, been relatively little crossover between the two approaches: one notable exception, perhaps, being the application of evolutionary computation to the evolution of sustainable institutions using Ostrom's ADICO grammar [12].

Moreover, although each approach has been applied to the same applications, neither alone seems to be sufficient to address the fundamental problem: how to *engineer* open, adaptive rule-base systems which remain sustainable over long periods of time and continue to fulfil their goals in the face of dynamic and unpredictable changes. On the one hand, the biologically-inspired approach tends to eschew, for example, the explicit representation of rules, social networking, conventional aspects of higher-order communication, and the creation of externalities through interactions and transactions. On the other hand, the socially-inspired approach, at least as represented by work on self-governing electronic institutions [8], tends to work mostly with fixed-size populations of fixed-ability agents operating in largely static environments.

However, in trying to find a ruleset that is congruent with the population and the environment, both for 'now' and 'in the future', there appears to be no 'one size fits all' optimum ruleset for all combinations of population, social network and environment; nor (given the contextual factors) is there a planning-type algorithm that can compute an 'ideal' ruleset for any particular combination of population, social network and environment. Therefore, these features of the problem suggest that recent advances in machine learning and evolutionary computation can provide the instruments for facilitating self-adaptation of a rule-based system over multiple timescales, if the appropriate computational framework could facilitate their integration. For this, though, we need first to identify the dimensions of flux, and their contextual factors, as addressed (respectively) in the next two sections.

3. The Dimensions of Flux

The driving force of the proposed research is the need to engineer systems that operate in dynamically changing environments. Therefore, we now outline the 'dimensions of flux', i.e. the features of an open complex network that can change, and for which appropriate adaptation mechanisms are required. These dimensions are broadly divided into four categories, with some sub-division. They are:

- the specification space, either:
  - by learning, i.e. coming to know which specification instance is most appropriate to meet or satisfice current operational conditions, and moving to that instance in time to be 'useful';
  - by innovation, i.e. by adding or deleting rules, or creating a completely new specification space (cf. often a consequence of revolution in social systems);
- the population:
  - the improvement of individual functional and/or reasoning capabilities, to being with,

1. This methodology was originally termed "sociologically" inspired computing, analogous to "biologically". However, inspiration can come from any of the social sciences: e.g. philosophy, psychology, politics, economics, etc. Hence here we will use the term "socially-inspired".

by training; in some systems, it might also be necessary to deal with decay (i.e. the deterioration of functional or reasoning capability);
  - evolution: change of population over time;

- the environment:
  - by natural causes;
  - by deliberate shaping (from terra-forming to cyber-forming [13]);

- the social network:
  - social learning and cultural evolution: for example the change of values over time and generations;
  - dynamic social psychology [14]: changes in the social network and structuration (the duality of agency and structure: structures are made out of agents, but agents have memory and knowledge about structures [15]), and also "rule-uration" (rules and processes are applied by and to agents; but agents have memory of rules and processes).

These dimensions of flux are inter-dependent, and note there is a kind of 'quantum' effect in the interaction of changeable rules with learning-capable components – changing the rules also changes the behaviour of the components, so the new rules no longer apply. Indeed, it is well-known in social systems that people do not adjust their behaviour in such a way as to simple comply with rules, but instead modify their behaviour to comply with incentives implied by the rules [16]. With this in mind, next section identifies six 'non-functional' contextual factors that also need to be considered when engineering systems in which change can occur along all the dimensions of flux.

4. Six 'Non-Functional' Contextual Factors

Given these four dimensions of flux in adaptive systems, we now highlight six non-functional factors that need to be taken into account. These factors are cost, fitness, training, fallibility, pace and externalities.

The first factor is that *computation costs resources*. Applying a process according to a set of rules comes with costs. In socio-economic systems, these are referred to as transaction costs. It has been observed that rules and processes work best when transaction costs are lowered, especially when 'piggy-backing' on normal behaviour. For example, Ostrom [1] observed that monitoring rules in irrigation systems were most effective when the farmers were able to observe each other in the transition from one supply to another. Similarly, Ober [17] observed that knowledge aggregation was most effective when citizens were given information to gossip about, since they were going to gossip anyway.

Costs are also impacted in path dependency [18], when the costs incurred by changing are greater than the perceived short-term benefits of changing, and the long-term benefits (which exceed the transaction costs) will be enjoyed by subsequent generations (e.g. climate change). Costs are particularly pertinent in computational systems with endogenous resources, and the cost of computation has to be 'paid for' from the very same resources that have to be distributed (e.g. CPU time, memory, battery power, etc.). There is no point, for example, in having an operational choice rule that computes the fairest distribution of resources, if it uses up all those resources in the computation.

Furthermore, it is essential to understand the relation between the rules and processes and the collective *values* that they are intended to serve. Crucially, the rules may have no apparent productivity but are critical to creating the value; in addition, they may be recognised as so critical to the process that they have value in and of themselves.

For example, in the study of jurisprudence, a fundamental question is which legal system is preferable: one which convicts all of the guilty and some of the innocent; or one which convicts only some of the guilty but none of the innocent. In deciding this question in favour of the latter, the UK legal system has developed a set of procedures, protocols, requirements and indeed rituals for trying court cases. These appear to be extremely time consuming and financially very costly – hence recent proposals for "fast track justice" which obviate these mechanisms. However, it is precisely these mechanisms that maintain a critical property of the legal system – i.e. that the innocent will not be convicted – and this turn gives the legal system its real value, at least in the view of the citizens affected by it. By contrast, "fast track justice" prioritises value in purely base financial terms and assumes a cheaper system is more 'preferable' to one that derives its value from principles of jurisprudence.

Related to cost and values, the second contextual factor is fitness (for purpose). This necessitates evaluating the degree to which the current specification instance (of the institution) is effective in satisficing the shared values of its members.

Note that this has to be an introspective process. One approach has been based on the idea of interactional justice, which supposes that; members have metrics for evaluating satisfaction at least according to their own experience; that there is an infrastructure that enables them to transform knowledge gained from personal experience to aggregated knowledge based on collective experience; and that there are mechanisms for 'reforming' the institution if it is considered to be falling short.

An additional complication is the trade-off between attention and the entropic tendency to oligarchy [19]. For example, it might again be costly if every agent participates in the metrication and the evaluation. So, some agents may perform the function, while other simply transmit the received wisdom. According to Rescher [20], one of the legitimate claims to consider in a system of distributive justice is the 'socially useful services', so the fitness evaluator may receive a larger proportion of resources thanks to performing this service for the community. However, this may breakdown, in two ways: one is that the evaluator believes the recompense is too little with regard to the service rendered and that the others are essentially free-riding; or that the evaluator may misrepresent the fitness, if the institution is generally unfit but personally beneficial. Herein lies a seed of oligarchy: lack of attention, lack of monitoring and motivation for personal gain.

The third contextual factor is training, if the population or membership changes over time. In many computational systems, it is assumed that new components enter 'fully fledged', as it were. However, this is generally not the case in either biological or social systems. Certainly in some social systems, there is a mismatch between newly-initiated members of a group, and the experience or long-serving members of the group. The long-serving members possess the knowledge, values and skills, and newcomers need to be initiated in these. In social settings, there are often various conventions, norms and rituals, some of which may seem pointless, but actually serve to coordinate expectations, provide a system of accountability between members, and create social capital in a relational economy [21] (and see below). According to Dewey [22], it is education that manages the transformation from uninitiated to initiated, and while this process also costs, it is arguably the key feature that has enabled sustainable common-pool resource management across multiple generations, i.e. those who were not originally party to the formation of the conventional rules, nevertheless proscribe their own behaviour accordingly.

The fourth factor to consider is the essential fallibility of democratic choice. It is possible for a process of preference selection to result in the wrong, or less favourable outcome. Ober [17] states that it was classical Athenian methods for organising useful knowledge that was at the root of the sustained and successful city-state, but even then wrong decisions were made (Ober cites the disastrous decision to invade Sicily in 411BC as the prelude to a period of oligarchic control before democracy was restored).

The recognition of fallibility, and the fact that rule changes can not always be tested in a double-blind randomised controlled trial in the same way as medicine (unless one starts with an autocratic state and not adverse to some negative consequences), is one of the motivating factors behind agent-based social simulation and legal methodologies such as ILTAM [23].

The fifth contextual factor is the pace of change. In some respects, selecting a specification instance is not unlike planning in a dynamic environment. If the rate of change is faster then the speed of planning computation, then if an agent starts re-planning after every perceived change in the environment, it may never complete a plan. On the other hand, it it proceeds to complete a plan regardless of changes in the environment, it may find that the plan is irrelevant to current state.

Therefore, the dilemma is that, on the one hand, reacting to every change with a change in the specification instance can be computationally costly and create instability; on the other hand, not reacting to a small change can result in a series of small changes not being reacted to, with a result that the specification instance ends up a 'long' way from the most desirable and the transactions costs in change are now excessive – the same problem of path dependency.

In fact, the pace of change seems to be at its worst when the rate of change of a sustainable resource synchronises with rate of change population: although the situation is worsening over generations, from the point of view of each generation the cognitive bias of "just noticeable difference" means that they accept the current situation as "normal" (it was ever thus) and so take action to sustain the resource.

The sixth contextual factor is externalities. Externalities in economics are consequences of commercial activities and transactions which affect third parties, without this being reflected in the price of the transaction. However, there are also benefits that arise from complying with rules, and this can have a significant affect on a social network. It is not just the link between two nodes that is important: it is also the strength and the nature of the link that has to be taken into account, as this can have a significant affect on individual decisions (e.g. forgiveness [24]) and collective action [21].

5. Self-Organising Electronic Institutions

The basic premise underlying self-organising electronic institutions was to apply the methodology of socially-inspired computing to Ostrom's design principles for self-governing institutions. This would *supply* institutions, i.e. as sets of rules, which encapsulated the principles in logical form: in [8], [25], the representation used was the Event Calculus (EC: [26]). Since such a specification can be its own program (i.e. executable specification), these EC specifications could be directly executed, hence *algorithmic self-governance*.

The precise details of the EC specification and operation are not essential for current purposes (see [6] for details). However, as an instance of the EC rules, consider the rule shown in Rule 1. This rule (axiom), states the designated agent C occupying the role of $chair$ is empowered (has the institutionalised power [27]) to declare the result of a vote on a motion M in the context of institution I, subject to certain conditions:

$$\begin{array}{l}
\mathbf{pow}(C, declare(C,W,M,I)) = true \quad \mathsf{holdsAt} \quad T \quad \leftarrow \\
\qquad role_of(C, head, I) = true \quad \mathsf{holdsAt} \quad T \quad \wedge \\
\qquad status(M,I) = closed \quad \mathsf{holdsAt} \quad T \quad \wedge \\
\qquad votes_cast(M,I) = X \quad \mathsf{holdsAt} \quad T \quad \wedge \\
\qquad eligible(M,I) = E \quad \mathsf{holdsAt} \quad T \quad \wedge \\
\qquad quora(I) = Q \quad \mathsf{holdsAt} \quad T \quad \wedge \\
\qquad quorate(X,E,Q) \quad \wedge \\
\qquad wdMethod(M,I) = WDM \quad \mathsf{holdsAt} \quad T \quad \wedge \\
\qquad winner_determination(WDM, X, W)
\end{array}$$

These conditions express that the winner should be declared in accordance with the way the votes were cast, whether or not the vote was quorate (enough of the electorate voted), and the winner determination method $wdMethod(M,I)$ for this institution.

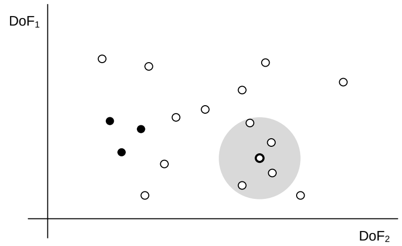

Figure 1. Specification Space with 2 DoF, distance, and invalid instances

Logically, the $quora$ and the $wdMethod$ are multi-valued fluents, i.e. they are propositions whose values can vary over time. For example, the quora could be simple majority, i.e. 50%, or in cases of major constitutional significance (like withdrawing from a major economic trading bloc and associated international treaties), at least 66%. Similarly, there are many different types of winner determination methods for selecting a choice from a set of expressed preferences: for example plurality, transferable vote, Borda count, etc.

Conceptually, these are variable parameters of the rule, or degrees of freedom in the dynamic norm-governed systems specification framework of Artikis [6]. In this framework, a set of rules R implicitly defines a specification space $\mathcal{L}$, where each instance of the specification space is characterised by a different assignment of values to each parameter in each rule. The size of this space is given by

$$\begin{aligned} \mid \mathcal{L} \mid \;\; = \;\; & (V_{1,1} \times V_{1,2} \times \ldots \times V_{1,P_1}) \times \\ & (V_{2,1} \times V_{2,2} \times \ldots \times V_{2,P_2}) \times \ldots \times \\ & (V_{R,1} \times V_{R,2} \times \ldots \times V_{R,P_R}) \end{aligned}$$

where $V_{i,j}$ is the number of values that the jth parameter of rule i can take, P_i is the number of parameters of rule i, and R is the number of rules in the set.

There are three points to note. Firstly, for any practical system, this space is likely to be "large". Secondly, there can also be rules about "moving" in the specification space: for example, certain configurations of parameter values may be considered unacceptable (invalid), or there may be constraints on how "far" the specification can be changed, based on a distance metric d (see Figure 1: this shows a specification with two degrees of freedom (unlikely), bold circle is the current specification instance, filled circles are invalid, allowable changes are within the gray area). Thirdly, it is possible to modify the morphology of the specification space altogether, for example by adding or deleting parameters, and even adding or deleting rules. For example, voting is not the only way to select one option from a set of alternatives.

The representation of institutional rules in axiomatic form is a form of knowledge codification, and maintaining codified knowledge is one of the knowledge management processes identified previously. Other works have addresses knowledge aggregation processes through opinion formation on dynamic social networks [14], knowledge alignment through voting protocols and social computational choice [28], and their interleaving [29].

However, while this provides an effective approach to self-organisation, e.g. by "moving" through the specification space, the open challenge is to engineer systems which can accommodate *all* the possible 'dimensions of flux' and the contextual factors. This, we argue, requires evolutionary computation and machine learning.

6. Bio-Inspired Computation

As just noted, the optimal parameter settings for a given rule depend heavily on the current environment, the existing population and the specific social network. As these three factors vary over time, the parameter settings must also vary. In the extreme, large shifts in the contextual space can result in an obsolete ruleset, that cannot be parameterised to achieve an acceptable level of performance. In this case, *new* rules are required.

Studies of biological systems provide numerous examples of systems that are able to adapt over multiple-time scales in order to maintain homeostasis (long-term stability) with finite resources. Machine-learning offers insights into systems that learn from data and adapt over time. Specifically, methods inspired by evolution and/or reinforcement learning have been successfully used to provide rapid parameter adaptation (i.e. *learning*) in collective systems, e.g. in swarm robotics. For example, evolutionary algorithms are used in [30] to adjust the parameters of a robot's ruleset during the course of a 'lifetime' to cope with a dynamically changing environment. [31] use a distributed algorithm based on an artificial-immune system algorithm (AIS) within a swarm robotics application to enable robots to adapt their individual foraging strategies over time based on available resources, environmental conditions and behaviours of other robots to maximise foraging.

While the above examples provide insight into how bio-inspired methods can be used to adapt systems on-the-fly to tune parameters to ensure congruence with an environment, it is also essential that a system can generate new knowledge when its current ruleset proves inadequate, i.e demonstates *innovation*. Recent developments in hyper-heuristics in an optimisation context have demonstrated that life-long optimisers can be designed that continuously perform both exploration (innovation) *and* exploitation (learning) [4]; the former discovers new optimal strategies for solving problems by evolving new semantic rules through genetic programming (long-term adaptation), whilst the latter continually refines existing strategies based on accumulated prior knowledge (rapid-adaptation). Others have used grammatical evolution, e.g. [32] to evolve rules for collective behaviours in robots or genetic programming to create rules for stock-trading [33].

Therefore, we propose that augmenting socially-inspired rule-based systems with methods to facilitate learning and innovation will provide a sound basis from which to engineer a new framework for engineering sustainable, distributed, autonomous systems. Biologically-inspired methods will provide the necessary machinery to operationalise Ostrom's principles that define the requirements that lead to sustainable behaviour, but without the necessary mechanisms to operationalise this. This brings about a paradigm-shift in the field of self-organising systems by providing a methodology to both *design* and *operate* long-lived, large-scale open systems in a principled manner.

7. Summary and Conclusions

In summary, we believe that open, adaptive systems of the kind that might be found in future ad hoc, opportunistic, green and sensor networks can be based on specifications of norm-governed systems. However, such systems can be in a constant state of flux along multiple dimensions, including the population, the social network, and the environment, as well as the rules themselves.

In this situation, the adaptable rules, on the one hand, interact in unexpected ways with an adaptive population, a dynamic network and an unpredictable environment on the other; with a number of other contextual factors thrown in. We have proposed the convergence of the socially- and biologically-inspired computing paradigms as the basis for a unifying framework for engineering such systems.

However, the acid test for such systems, whether they are really long-standing, is whether such systems can really occur with drastic, unanticipated, 'surprise' events. The key requirement is not that there should be be no loss in performance, but that in the aftermath of the surprise, operation should recover to achieve the same pre-surprise levels of performance.

Acknowledgments

The first author has been partially supported by the Leverhulme Trust, Research Fellowship RF-2016-451. Thanks to the anonymous reviewers for many helpful comments.

References

[1] E. Ostrom, *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, 1990.

[2] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[3] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Springer, 2003.

[4] K. Sim, E. Hart, and B. Paechter, "A lifelong learning hyper-heuristic method for bin packing," *Evolutionary Computation*, vol. 23, no. 1, pp. 37–67, 2015.

[5] P. Andrews, F. Polack, A. Sampson, S. Stepney, and J. Timmis, "The cosmos process, version 0.1: A process for the modelling and simulation of complex systems," University of York, Technical Report YCS-2010-453, 2010.

[6] A. Artikis, "Dynamic specification of open agent systems," *Journal of Logic and Computation*, vol. 22, no. 6, pp. 1301–1334, 2012.

[7] A. García-Camino, P. Noriega, and J. A. Rodríguez-Aguilar, "Implementing norms in electronic institutions," in *AAMAS*, 2005, pp. 667–673.

[8] J. Pitt, J. Schaumeier, and A. Artikis, "Axiomatisation of socio-economic principles for self-organising institutions: Concepts, experiments and challenges," *TAAS*, vol. 7, no. 4, pp. 39:1–39:39, 2012.

[9] A. Jones, A. Artikis, and J. Pitt, "The design of intelligent socio-technical systems," *Artificial Intelligence Review*, vol. 39, no. 1, pp. 5–20, 2013.

[10] E. Hart and K. Sim, "On constructing ensembles for combinatorial optimisation," *Evolutionary Computation*, vol. online, 2017.

[11] J. Pitt, D. Busquets, A. Bourazeri, and P. Petruzzi, "Collective intelligence and algorithmic governance of socio-technical systems," in *Social Collective Intelligence*, D. Miorandi, V. Maltese, M. Rovatsos, A. Nijholt, and J. Stewart, Eds. Springer, 2014, pp. 31–50.

[12] A. Ghorbani, G. Bravo, U. Frey, and I. Theesfeld, "Self-organization in the commons : An empirically-tested model," *Environmental Modelling & Software*, vol. 96, no. 30–45, 2017.

[13] J. Bradshaw, N. Suri, A. Canas, R. Davis, K. Ford, R. Hoffman, R. Jeffers, and T. Reichherzer, "Terraforming cyberspace," *Computer*, vol. 34, no. 7, 2001.

[14] R. Vallacher, S. Read, and A. Nowak, Eds., *Computational Social Psychology*. Psychology Press, 2017.

[15] A. Giddens, *The constitution of society: Outline of the theory of structuration*. Cambridge, UK: Polity Press, 1984.

[16] E. López, *The Pursuit of Justice: Law and Economics of Legal Institutions*. New York, NY: Palgrave MacMillan, 2010.

[17] J. Ober, *Democracy and Knowledge*. Princeton Univ. Press, 2008.

[18] R. Collier and D. Collier, *Shaping The Political Arena*. Princeton University Press, 1991.

[19] R. Michels, *Political Parties: A Sociological Study of the Oligarchical Tendencies of Modern Democracy*. Transaction Publ., 2009 [1911].

[20] N. Rescher, *Distributive Justice*. Bobbs-Merrill, 1966.

[21] P. E. Petruzzi, J. Pitt, and D. Busquets, "Inter-institutional social capital for self-organising 'nested enterprises'," in *10th IEEE SASO*, 2016, pp. 90–99.

[22] J. Dewey, *Democracy and Education: An Introduction to the Philosophy of Education*. New York: Macmillan, 1916.

[23] A. Seidman and R. B. Seidman, "Iltam: Drafting evidence-based legislation for democratic social change," *BUL Rev.*, vol. 89, p. 435, 2009.

[24] A. Vasalou, A. Hopfensitz, and J. Pitt, "In praise of forgiveness: Ways for repairing trust breakdowns in one-off online interactions," *Int. J. Hum.-Comput. Stud.*, vol. 66, no. 6, pp. 466–480, 2008.

[25] J. Pitt, D. Busquets, and S. Macbeth, "Distributive justice for self-organised common-pool resource management," *ACM Trans. Auton. Adapt. Syst.*, vol. 9, no. 3, p. 14, 2014.

[26] R. Kowalski and M. Sergot, "A logic-based calculus of events," *New Generation Computing*, vol. 4, pp. 67–95, 1986.

[27] A. Jones and M. Sergot, "A formal characterisation of institutionalised power," *Journal of the IGPL*, vol. 4, no. 3, pp. 427–443, 1996.

[28] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet, "A short introduction to computational social choice," in *Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, ser. LNCS, no. 4362. Springer, 2007, pp. 51–69.

[29] J. Pitt, D. Ramirez-Cano, M. Draief, and A. Artikis, "Interleaving multi-agent systems and social networks for organized adaptation," *CMOT*, vol. 17, no. 4, pp. 344–378, 2011.

[30] A. Steyven, E. Hart, and B. Paechter, "An investigation of environmental influence on the benefits of adaptation mechanisms in evolutionary swarm robotics," in *GECCO*, 2017, pp. 155–162.

[31] N. Capodieci, E. Hart, and G. Cabri, "Artificial immunology for collective adaptive systems design and implementation," *ACM Trans. Auton. Adapt. Syst.*, vol. 11, no. 2, pp. 6:1–6:25, Jun. 2016.

[32] E. Ferrante et al., "GESwarm: Grammatical evolution for the automatic synthesis of collective behaviors in swarm robotics," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, 2013.

[33] Y. Hu et al., "Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review," *Applied Soft Computing*, vol. 36, pp. 534–551, 2015.